FAST FUNCTIONAL INTEGRALS WITH APPLICATION TO DIFFERENTIAL EQUATIONS MODELS

JOHN TILLINGHAST

1. Abstract

A new method is introduced which uses higher-order Laplace approximation to evaluate functional integrals much faster than existing methods. An implementation in MATLAB is called SLAM-FIT (Sparse Laplace Approximation Method for Functional Integration on Time) or simply SLAM. In this paper SLAM is applied to estimate parameters of mixed models that require functional integration. It is compared with two more general packages which can be used to do functional integration. One is Stan, a recent and very general package for integrating and estimating using hybrid Monte Carlo. The other is INLA, a recent R package which uses Laplace approximations for Gaussian Markov random fields. In both cases it is able to get near-identical or equivalent results in less time for moderately sized data sets. The fundamental speed advantage of the algorithm may be greater than it appears, because SLAM is running in pure MATLAB while the other two packages use optimized compiled code.

Keywords: compartment models, predator-prey, SIR model, path integral, functional integral, Laplace approximation, saddlepoint method.

2. Introduction

This paper has two purposes: to introduce a new, much faster computational method for functional integration, and to show its usefulness in estimating parameters of differential equation models. For a time-dependent random process, a functional integral can be thought of as the integral over all possible values at all times over the relevant time interval [Dirac, 1933, Feynman and Hibbs, 2010, Schulman, 2005, Kleinert, 2009]. Technically they are integrals over a function space such as a space of Brownian motion paths. They have been used for at least eighty years in many branches of science and applied mathematics including quantum and statistical mechanics, polymer science, probability, and finance. Typically they are used to compute the chance of a system evolving from one state to a different one: any in-between path is possible, but the probability is given by the integral over the possible paths. This makes them a natural tool to use for analyzing systems which are imperfectly modeled by ordinary differential equations. In life science there are predator-prey models [Edelstein-Keshet, 1988], infectious disease models [Kermack and McKendrick, 1927, Anderson and May, 1991], pharmacokinetics models [Gelman et al., 1996], and many others. Most of macroeconomics depends on differential equation-based

models such as the Solow growth model or the ISLM model [Romer, 2011]. These models can be made more realistic by considering noise or stochastic behavior.

A major goal of differential equations models is to estimate system parameters, such as how infectious a virus is, or reaction rates for chemicals and enzymes. Traditional approaches involve simplified models on transformed variables [Lineweaver and Burk, 1934] or optimizing some measure of data fit over exact solutions to the ODEs [Anderson and May, 1991]. Recent methods use a generalized smoothing approach. For a finite-dimensional basis, such as splines, it is possible to approximately fit both the differential equations and the data [Ramsay J. et al., 2007, Campbell and Steele, 2012]. As with smoothing, the trade-off between data fit and ODE fit can be chosen by cross-validation or by a mixed-model approach.

In this paper, we use a very different approach: we discretize time, and model the underlying true values as a first-order Markov process. Then we assume that the data were generated with a distribution depending on the true value. Not all of the time points have data; in fact, it may be important to add in-between time points, just as they would be needed for a finite difference method. Then we define a marginalized pseudolikelihood of the system parameters as an integral over the possible values which the variables might have taken at the time points. This approximates a functional integral over a continuous random process.

There are both analytic and computational methods for evaluating functional integrals. Analytic methods require a tractable integrand or a good approximation to one. Very clever schemes have been painstakingly developed to transform functional integrals and make them tractable [Kleinert, 2009, Schulman, 2005]. Far more problems have so far required time-consuming Monte Carlo methods.

Laplace and saddlepoint approximations have also been used to approximate analytic functional integrals when possible. But this appears to be the first time that the approximations have been used for numerical evaluation. One reason for this may be that some of the higher-order terms, involving third- and fourth-order tensors, are difficult to evaluate efficiently.

This paper introduces a way to compute the higher-order terms quickly for numerical functional integrals. This requires computing a critical path (a unique most likely realization) and expanding the log-likelihood around it. The log-likelihood is a sum of interactions between neighboring time points. Consequently the relevant Hessian and the tensors of third- and fourth-order derivatives are sparse with a simple block structure: all entries corresponding to non-neighboring time points are zero. By using this sparsity structure, it is possible to calculate the higher-order terms in $O(np^4)$ time, where n is the number of time points and p is the number of variables in the system.

A variance-stabilizing transformation is needed when the data times are separated by many in-between time steps. In general it makes the integrals more accurate than when untransformed. SLAM can be used with the basic Laplace term alone, or with the higher-order terms. Higher-order terms make the integrals more accurate, but for the examples tested, the parameter estimates are close to those found using the basic Laplace approximation.

To evaluate speed and accuracy, SLAM was compared with two popular systems for mixed models. Results for an infectious disease data set are equivalent to results from Stan [STAN Development Team], the leading general-purpose Monte Carlo system, but SLAM is much faster. There is also a well-implemented Laplace-based method, INLA [Rue et al., 2009, Rue et al.], designed for hierarchical models and Gaussian Markov random fields.

For random fields, INLA takes advantage of sparsity, but the higher-order terms still take quadratic time. The special approximation introduced here can only be applied to GMRFs on a line, not for other type covered by INLA. To compare INLA and SLAM, we used an INLA demonstration data set (Tokyo rainfall by day of year). For repeats of the Tokyo data, estimates are well within the CI for the two methods. INLA (with full higher-order terms) is faster at 1100 data points but SLAM is faster beyond that. INLA using simplified higher-order terms remains somewhat faster up to the largest set tested (about 15000 data points).

3. METHODS

3.1. **Markov process setup.** The model has two stages: a Markov process for the true values, and on top of that, measurements with noise. We assume that the Markov and measurement likelihoods depend on parameters $\boldsymbol{\theta}$. We show how to define a marginalized likelihood for $\boldsymbol{\theta}$. Maximizing the marginalized likelihood gives an estimate of $\boldsymbol{\theta}$.

In this paper we will use Latin indexes ("i") to denote time points and matrix blocks. We use Greek indexes ("α") to denote individual vector and matrix entries. Each time point has as one entry for every variable. This becomes important when analyzing the blocking structure. We also use the summation convention for duplicated indexes (e.g. $v_\gamma = A_{\alpha\beta}T_{\alpha\beta\gamma}$ means $v_\gamma = \sum_{\alpha,\beta} A_{\alpha\beta}T_{\alpha\beta\gamma}$.)

The Markov process has a vector value at each time point, denoted by $\mathbf{y}_i$ at step i (time t_i). (We use $\mathbf{y}$ without an index to mean the entire realization, i.e., the concatenation of the $\mathbf{y}_i$.) The system also has parameters $\boldsymbol{\theta}$ that involved in the Markov process or the measurement error. From the Markov process, at time t_{i+1} there is a conditional probability density for $\mathbf{y}_{i+1}$, given the value of $\mathbf{y}_i$:

$$f_{trans}\left(\mathbf{y}_{i+1}|\mathbf{y}_i,\boldsymbol{\theta}\right) = exp\big(-\ell_{trans}\left(\mathbf{y}_{i+1}|\mathbf{y}_i,\boldsymbol{\theta}\right)\big)$$

so given the initial values $\mathbf{y}_1$, we have the overall conditional density for a realization $\mathbf{y}$ at all time points:

$$f_{dyn}\left(\mathbf{y}|\mathbf{y}_1,\boldsymbol{\theta}\right) = \prod_{i=1}^{n-1} f_{trans}\left(\mathbf{y}_{i+1}|\mathbf{y}_i,\boldsymbol{\theta}\right)$$

If we have a prior $\pi(\mathbf{y}_1)$ we can define the probability of a realization:

$$f_{dyn}(\mathbf{y}|\boldsymbol{\theta}) = \pi(\mathbf{y}_1) f_{dyn}\left(\mathbf{y}|\mathbf{y}_1,\boldsymbol{\theta}\right)$$

Some of the time points have data. Let I_{data} be the set of their indexes. Call the data values $\mathbf{y}_i^*$ for $i \in I_{data}$. We assume that the data are independently generated with a probability density depending on the true value and parameters:

$$f_{data}\left(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}\right) = exp\big(-\ell_{data}\left(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}\right)\big)$$

Depending on the system, the error could be measurement error from an instrument, or from sampling error, or from some other source. The overall density of the given realization and data is

$$f(\mathbf{y}, \mathbf{y}^* | \boldsymbol{\theta}) = \pi(\mathbf{y_1}) \prod_{i=1}^{n-1} f_{dyn}\left(\mathbf{y}_{i+1} | \mathbf{y}_i, \boldsymbol{\theta}\right) \prod_{i \in I_{data}} f_{data}(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta})$$

This means that the pdf of $\mathbf{y}^*$ given $\boldsymbol{\theta}$ is given by

$$\begin{aligned} f\left(\mathbf{y}^* | \boldsymbol{\theta}\right) &= \int_{\mathbb{R}^N} \pi(\mathbf{y_1}) \prod_{i=1}^{n-1} f_{dyn}\left(\mathbf{y}_{i+1} | \mathbf{y}_i, \boldsymbol{\theta}\right) \prod_{i \in I_{data}} f_{data}(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}) d^N\mathbf{y} \\ &= \exp\left(log\ \pi(\mathbf{y_1}) - \sum_{i \in I_{data}} \ell_{data}\left(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}\right) - \sum_{i=1}^{n-1} \ell_{trans}\left(\mathbf{y}_{i+1} | \mathbf{y}_i, \boldsymbol{\theta}\right)\right) \end{aligned}$$

Given $\pi(\mathbf{y_1})$ it is possible to define a marginal likelihood for $\boldsymbol{\theta}$ by integrating $f\left(\mathbf{y}^* | \boldsymbol{\theta}\right)$ over the latent true values. But in general, we don't know $\pi(\mathbf{y_1})$, and we use the improper prior $\pi(\mathbf{y_1}) = 1$. The integral is finite because of the multiplication by f_{data} which can be thought of as a prior on the data points. This gives us the marginalized pseudolikelihood

$$\begin{aligned} M\left(\boldsymbol{\theta}\right) &= \int_{\mathbb{R}^N} \prod_{i=1}^{n-1} f_{dyn}\left(\mathbf{y}_{i+1} | \mathbf{y}_i, \boldsymbol{\theta}\right) \prod_{i \in I_{data}} f_{data}(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}) d^N\mathbf{y} \\ &= \int_{\mathbb{R}^N} e^{-\ell(\mathbf{y})} d^N\mathbf{y} \end{aligned}$$

where

$$\ell(\mathbf{y}) = \sum_{i \in I_{data}} \ell_{data}(\mathbf{y}_i^* | \mathbf{y}_i, \boldsymbol{\theta}) + \sum_{1}^{N-1} \ell_{trans}\left(\mathbf{y}_{i+1} | \mathbf{y}_i, \boldsymbol{\theta}\right).$$

This marginalized pseudolikelihood is what we will maximize to estimate $\boldsymbol{\theta}$.

3.2. **Laplace approximation (basic and higher-order).** In this subsection, we look closely at how to get a good approximation for M. In addition, assume that $\ell(\mathbf{y})$ has a single peak at $\hat{\mathbf{y}}$. Then we can expand around $\hat{\mathbf{y}}$:

$$
\begin{aligned}
\mathbf{y} &= \hat{\mathbf{y}} + \boldsymbol{\varepsilon} \\
M &= \int_{\mathbb{R}^N} e^{-\ell(\mathbf{y})} d^N \mathbf{y} \\
&= \int_{\mathbb{R}^N} e^{-\ell(\hat{\mathbf{y}}+\boldsymbol{\varepsilon})} d^N \boldsymbol{\varepsilon} \\
&= \int_{\mathbb{R}^N} \exp\left(-\ell\left(\hat{\mathbf{y}}\right) - \frac{1}{2}\ell^{(2)}_{\alpha\beta}\left(\hat{\mathbf{y}}\right)\varepsilon_\alpha\varepsilon_\beta - \frac{1}{3!}\ell^{(3)}_{\alpha\beta\gamma}\left(\hat{\mathbf{y}}\right)\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma - \frac{1}{4!}\ell^{(4)}_{\alpha\beta\gamma\lambda}\left(\hat{\mathbf{y}}\right) - ... \right) d^N\boldsymbol{\varepsilon} \\
&= e^{-\ell(\hat{\mathbf{y}})} \int_{\mathbb{R}^N} e^{-\frac{1}{2}H_{\alpha\beta}\varepsilon_\alpha\varepsilon_\beta} \exp\left(-\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma - \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda - ... \right) d^N\boldsymbol{\varepsilon}
\end{aligned}
$$

where $\mathbf{H}$ is the Hessian of $\ell(\mathbf{y})$ at $\hat{\mathbf{y}}$, and $\mathbf{T}$ and $\mathbf{F}$ are the tensors of third and fourth derivatives at $\hat{\mathbf{y}}$.

We can think of $\boldsymbol{\varepsilon}$ as a Gaussian random variable, with precision matrix $\mathbf{H}$, and this integral is the expectation of a function of $\boldsymbol{\varepsilon}$:

$$
M = e^{-\ell(\hat{\mathbf{y}})}(2\pi)^{N/2}|\mathbf{H}|^{-1/2} E\left\{ \exp\left(-\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma - \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda - ... \right) \right\}
$$

The expectation can be expanded and then approximated with an asymptotic series.

$$
\begin{aligned}
M &\propto E\left\{ \sum_{r=0}^{\infty} (-)^r \frac{1}{r!} \left(\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma + \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda + ... \right)^r \right\} \\
&= E\left\{ 1 - \left(\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma + \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda + ... \right) \right\} \\
&\qquad + \frac{1}{2!}\left(\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma + \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda + ... \right)^2 + ... \Bigg\} \\
&\sim 1 - E\left(\frac{1}{3!}T_{\alpha\beta\gamma}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma + \frac{1}{4!}F_{\alpha\beta\gamma\lambda}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda + ... \right) \\
&\qquad + \frac{1}{2!}E\left(\left(\frac{1}{3!} \right)^2 T_{\alpha\beta\gamma}T_{\lambda\mu\nu}\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda\varepsilon_\mu\varepsilon_\nu + ... \right) - ... \\
&\sim 1 - \frac{1}{4!}F_{\alpha\beta\gamma\lambda}E\left(\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda\right) + \frac{1}{2!}\left(\frac{1}{3!} \right)^2 T_{\alpha\beta\gamma}T_{\lambda\mu\nu}E\left(\varepsilon_\alpha\varepsilon_\beta\varepsilon_\gamma\varepsilon_\lambda\varepsilon_\mu\varepsilon_\nu\right) + ...
\end{aligned}
$$

This series diverges, but the terms shown can produce a good approximation to M on their own. A similar, often better approximation is given by a cumulant expansion for *log* M [Shun and McCullagh, 1995, McCullagh, 1987], where the first higher-order terms are the same as above, but the product is turned into a sum:

$$\begin{aligned} log\ M \sim & -\ell\left(\hat{\mathbf{y}}\right) + \frac{N}{2} log\left(2\pi\right) - \frac{1}{2} log|\mathbf{H}| \\ & - \frac{1}{4!} F_{\alpha\beta\gamma\lambda} E\left(\varepsilon_\alpha \varepsilon_\beta \varepsilon_\gamma \varepsilon_\lambda\right) \\ & + \frac{1}{2}\left(\frac{1}{3!}\right)^2 T_{\alpha\beta\gamma} T_{\lambda\mu\nu} E\left(\varepsilon_\alpha \varepsilon_\beta \varepsilon_\gamma \varepsilon_\lambda \varepsilon_\mu \varepsilon_\nu\right) \\ & + ... \\ = & -\ell\left(\hat{\mathbf{y}}\right) + \frac{N}{2} log\left(2\pi\right) - \frac{1}{2} log|\mathbf{H}| + IV + IIIa + IIIb + ... \end{aligned}$$

where

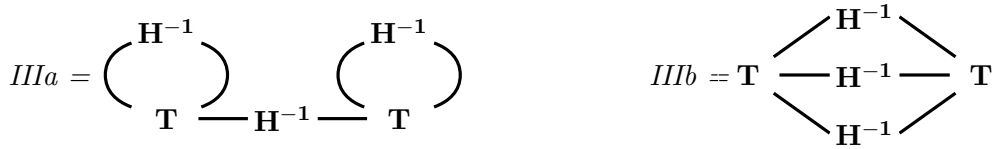
$$\begin{aligned} IV &= -\frac{1}{24} \cdot 3 F_{\alpha\beta\gamma\lambda} H^{-1}_{\alpha\beta} H^{-1}_{\gamma\lambda} \\ IIIa &= \frac{1}{72} \cdot 9 H^{-1}_{\alpha\beta} T_{\alpha\beta\gamma} H^{-1}_{\gamma\lambda} T_{\lambda\mu\nu} H^{-1}_{\mu\nu} \\ IIIb &= \frac{1}{72} \cdot 6 T_{\alpha\beta\gamma} H^{-1}_{\alpha\lambda} H^{-1}_{\beta\mu} H^{-1}_{\gamma\nu} T_{\lambda\mu\nu} \end{aligned}$$

We use the cumulant expansion, for $log\ M$, because the derivatives are simpler and because in many cases it is more accurate [Shun and McCullagh, 1995].

Terms IIIa and IIIb involve the same tensors, but the sums are very different. In term IIIa, as long as we have the near-diagonal entries of $\mathbf{H}^{-1}$, we can immediately convert the third-order tensors into vectors $v_\gamma = H^{-1}_{\alpha\beta} T_{\alpha\beta\gamma}$. Then $IIIa = \mathbf{v^T H^{-1} v}$. In section 4 we explain how to compute the near-diagonal entries in $O(np^3)$ time [Asif and Moura, 2005]; getting $\mathbf{v^T H^{-1} v}$ is even faster because $\mathbf{H}$ is block-tridiagonal.

Term IIIb is different and much more difficult. Each $\mathbf{H^{-1}}$ connects one mode of the first $\mathbf{T}$ to a mode of the second $\mathbf{T}$, and $\mathbf{H^{-1}}$ is full. This means that entries from one time point of the first $\mathbf{T}$ are multiplied by entries from all other time points in the second $\mathbf{T}$, not just entries from the neighboring time points. This would seem to mean that IIIb requires quadratic time (in n) to compute. The appendix shows how it can be done in linear time by using power series and recurrence relations.

Graphically, the difference can be represented this way:



Here each line represents a contraction over two modes: one from each tensor, if there are two, or two from a single tensor. This notation will be helpful later when doing more complicated manipulations on term IIIb.

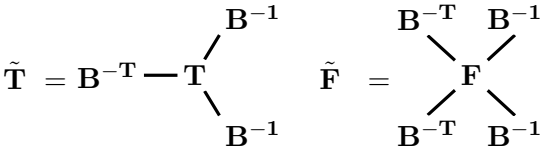
It is worth emphasizing that all three of these terms are invariant under linear transformations. If we make a change of variables $\tilde{\mathbf{y}} = \mathbf{B}\mathbf{y}$, then

$$\frac{\partial}{\partial \tilde{\mathbf{y}}} = \mathbf{B}^{-1}\frac{\partial}{\partial \mathbf{y}}$$

and the derivatives tensors (at the critical path) become

$$\begin{aligned}
\tilde{H}_{\alpha\beta} = \frac{\partial^2 \ell}{\partial \tilde{y}_\alpha \partial \tilde{y}_\beta} &= H_{\mu\nu} B^{-1}_{\mu\alpha} B^{-1}_{\nu\beta} && \tilde{\mathbf{H}} = \mathbf{B^{-T}} \mathbf{H} \mathbf{B^{-1}} \\
\tilde{T}_{\alpha\beta\gamma} &= T_{\mu\nu\rho} B^{-1}_{\mu\alpha} B^{-1}_{\nu\beta} B^{-1}_{\rho\gamma} && \tilde{\mathbf{T}} = (\mathbf{B^{-T}} \otimes \mathbf{B^{-T}} \otimes \mathbf{B^{-T}})(\mathbf{T}) \\
\tilde{F}_{\alpha\beta\gamma\lambda} &= F_{\mu\nu\rho\sigma} B^{-1}_{\mu\alpha} B^{-1}_{\nu\beta} B^{-1}_{\rho\gamma} B^{-1}_{\sigma\lambda} && \tilde{\mathbf{F}} = (\mathbf{B^{-T}} \otimes \mathbf{B^{-T}} \otimes \mathbf{B^{-T}} \otimes \mathbf{B^{-T}})(\mathbf{F})
\end{aligned}$$

Then the sum $-\frac{1}{24}\tilde{F}_{\alpha\beta\gamma\lambda}\tilde{H}^{-1}_{\alpha\beta}\tilde{H}^{-1}_{\gamma\lambda}$ is still equal to term IV, $\frac{9}{72}\tilde{H}^{-1}_{\alpha\beta}\tilde{T}_{\alpha\beta\gamma}\tilde{H}^{-1}_{\gamma\lambda}\tilde{T}_{\lambda\mu\nu}\tilde{H}^{-1}_{\mu\nu}$ is still term IIIa, etc. In the simplified matrix and graphical notation, the transformation of $\mathbf{T}$ and $\mathbf{F}$ can be written as



In reality, both tensors are being multiplied symmetrically in all modes. The reason for using transposes is because if a matrix is on the left side, its rows are multiplied, and if a matrix is on the right side, its columns are multiplied. This is consistent with the usual direction of matrix multiplication. Using the same convention will make things simpler in the Appendix.

4. The Sparsity Structure and How to Use It

Take another look at the log likelihood $\ell(\mathbf{y})$ (3.1). Each term in ℓ_{data} uses values from one time point. Each term in ℓ_{dyn} uses values from two adjacent time points.

This means that if i is any time point, the blocks $\mathbf{H_{i,i}}$, $\mathbf{H_{i,i+1}}$, and $\mathbf{H_{i,i-1}}$ can be nonzero. For any other j, i.e. if $|i-j|>1$, $\mathbf{H_{i,j}} = \frac{\partial^2 \ell}{\partial \mathbf{y}_i \partial \mathbf{y}_j} = \mathbf{0_{p\times p}}$. Such a matrix is called block-tridiagonal (see figure 2).

Likewise, $\mathbf{T_{i,j,k}}$ is zero unless $\max(|i-j|,|j-k|,|i-k|) \le 1$. And $\mathbf{F_{i,j,k,l}}$ is zero if any of the $\binom{4}{2} = 6$ differences is greater than 1.

4.1. Setup, block-diagonals, block off-diagonals, levels.

In this subsection we see how to work with $\mathbf{H}$, then apply that to computing IV and $IIIa$. The details for $IIIb$ are worked out in the appendix.

If $\hat{\mathbf{y}}$ is a true local minimum of ℓ, then $\mathbf{H}$ has to be positive definite. This is crucial, because we need the Cholesky decomposition for the calculation.

Here $\mathbf{D}$ is block-diagonal but not symmetric. In fact, $\mathbf{D}$ is lower-triangular. $\mathbf{A}$ is block off-diagonal, level -1. This means that $\mathbf{A_{i,j}}$ can only be nonzero if $i = j+1$.

$$
\begin{aligned}
\mathbf{H} &= \begin{pmatrix} \mathbf{P_1} & \mathbf{Q_1}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{Q_1} & \mathbf{P_2} & \mathbf{Q_2}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_2} & \mathbf{P_3} & \mathbf{Q_3}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{Q_3} & \mathbf{P_4} \end{pmatrix} \\
&= \mathbf{L}\mathbf{L}^\mathbf{T}, \text{where} \\
\mathbf{L} &= \begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{E_1} & \mathbf{D_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E_2} & \mathbf{D_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E_3} & \mathbf{D_4} \end{pmatrix} \\
&= \mathbf{D} + \mathbf{E} \\
&= \mathbf{D}\big(\mathbf{I} + \mathbf{D^{-1}E}\big) \\
&= \mathbf{D}\big(\mathbf{I} - \mathbf{A}\big)
\end{aligned}
$$

FIGURE 1. Decomposition of $\mathbf{H}$.

If $\mathbf{X}$ has level $l_X = 1$, and $\mathbf{Y}$ has level $l_Y = -2$, then $\mathbf{XY}$ has level $l = l_X + l_Y = -1$. The third block of level $\mathbf{XY}$ is zero because $\mathbf{Y}$ has only 2 nonzero blocks.

$$
\begin{aligned}
\mathbf{XY} &= \begin{pmatrix} \mathbf{0} & \mathbf{X_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{X_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{X_3} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{Y_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_2} & \mathbf{0} & \mathbf{0} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{X_2Y_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X_3Y_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}
\end{aligned}
$$

FIGURE 2. Off-diagonal matrix multiplication

In general, the $np \times np$ matrix $\mathbf{X}$ is block off-diagonal, level l, if the blocks $\mathbf{X_{i,j}} \neq \mathbf{0}$ only when $j = i + l$.

Block-off-diagonals have a raising and lowering property, demonstrated in Figure 2. If $\mathbf{X}$ is block off-diagonal with level l_X, and $\mathbf{Y}$ is block off-diagonal with level l_Y, then $\mathbf{XY}$ is block off-diagonal with level $l_X + l_Y$. "Raising the level by 1" happens when you multiply on either size by a block off-diagonal matrix of level 1. "Lowering by 1" is the same as "raising by -1", which means multiplying by a block off-diagonal of level -1.

4.2. **Computing near-diagonal elements of $\mathbf{H^{-1}}$.** Moving on with the computation,

$$\begin{aligned}\mathbf{H} &= \mathbf{D}\big(\mathbf{I}-\mathbf{A}\big)\big(\mathbf{I}-\mathbf{A}\big)^{\mathbf{T}}\mathbf{D}^{\mathbf{T}}\\ \mathbf{H^{-1}} &= \mathbf{D^{-T}}\big(\mathbf{I}-\mathbf{A}\big)^{\mathbf{-T}}\big(\mathbf{I}-\mathbf{A}\big)^{\mathbf{-1}}\mathbf{D^{-1}}\end{aligned}$$

The power series for $(\mathbf{I}-\mathbf{A})^{-1}$ actually terminates because $\mathbf{A}$ is strictly lower-triangular. So we can expand and group terms:

$$\begin{aligned}\mathbf{H} &= \mathbf{D^{-T}}\Big(\sum_{q\geq 0}\mathbf{A^{Tq}}\Big)\Big(\sum_{r\geq 0}\mathbf{A^{r}}\Big)\mathbf{D^{-1}}\\ &= \mathbf{D^{-T}}\Big(\sum_{q\geq 0}\sum_{r\geq 0}\mathbf{A^{Tq}A^{r}}\Big)\mathbf{D^{-1}}\end{aligned}$$

The terms of the double sum can be grouped by level so as to give us the near-diagonal levels of $\mathbf{H^{-1}}$. This results in an algorithm similar to the block-tridiagonal case of [Asif and Moura, 2005].

Each term in the double sum has exactly one level, $q-r$. Now we will group the terms by level. Break the sum into the two cases, $q<r$ and $q\geq r$:

$$\begin{aligned}\sum_{q\geq 0}\sum_{r\geq 0}\mathbf{A^{Tq}A^{r}} &= \sum_{r>q\geq 0}\mathbf{A^{Tq}A^{r}} + \sum_{q\geq r\geq 0}\mathbf{A^{Tq}A^{r}}\\ &= \sum_{\mu>0,q\geq 0}\mathbf{A^{Tq}A^{q+\mu}} + \sum_{\nu\geq 0,r\geq 0}\mathbf{A^{T(r+\nu)}A^{r}}\\ &= \sum_{\mu>0}\Big(\sum_{q\geq 0}\mathbf{A^{Tq}A}^{q}\mathbf{A^{\mu}}\Big) + \sum_{\nu>0}\Big(\sum_{r\geq 0}\mathbf{A^{T\nu}A^{Tr}A^{r}}\Big)\\ &= \sum_{\mu>0}\mathbf{SA^{\mu}} + \mathbf{S} + \sum_{\nu>0}\mathbf{A^{T\nu}S}\end{aligned}$$

where

$$\mathbf{S} = \sum_{q\geq 0}\mathbf{A^{Tq}A^{q}}.$$

Now we will show how to compute $\mathbf{S}$ in $O(np^3)$ time by a block matrix recurrence relation.

Obviously,

$$\mathbf{S} = \mathbf{I} + \mathbf{A^{T}SA}$$

The map $S \to \mathbf{A^{T}SA}$ has a special property: each block of $\mathbf{A^{T}SA}$ only depends on the *following* block of $\mathbf{S}$, and the last block of $\mathbf{A^{T}SA}$ is zero.

$$\begin{pmatrix} \mathbf{0} & \mathbf{A_1}^T & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_2}^T & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A_{n-1}^T} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{S_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S_{n-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{S_n} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{A_2} & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A_{n-1}} & \mathbf{0} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{A_1^T S_2 A_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_2^T S_3 A_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A_{n-1}^T S_n A_{n-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Because of this property, we can solve $\mathbf{S = I + A^T S A}$ by a reverse iteration:

$$\mathbf{S_n = I}$$
$$\mathbf{S_k = I + A_k^T S_{k+1} A_k}$$

for k going from $n-1$ down to 1.

This gives a total of $O(n)$ block multiplications, each of which takes up to $O(p^3)$ flops for total complexity $O(np^3)$. There may be special cases when there are less than $O(p^3)$ flops per block, but that would be somewhat unusual because $\mathbf{A}$ is generated by a Cholesky decomposition.

Having computed $\mathbf{S}$, we now have

$$\mathbf{H^{-1} = D^{-T}}\Big(\sum_{\mu>0} \mathbf{S A^{\mu}} + \mathbf{S} + \sum_{\nu>0} \mathbf{A^{T\nu} S}\Big)\mathbf{D^{-1}}$$

We will never actually compute all the entries of $\mathbf{H^{-1}}$ – it is full and would need too many flops. For terms IV and IIIa, we only need the block-tridiagonal part,

$$\mathbf{D^{-T}}\Big(\mathbf{SA} + \mathbf{S} + \mathbf{A^T S}\Big)\mathbf{D^{-1}},$$

which has $O(np^2)$ entries and takes $O(np^3)$ time to compute.

4.3. **Computing IV and IIIa.** Term IV is simple: each nonzero element of $\mathbf{F}$ occurs exactly once, and is multiplied by block-tridiagonal elements of $\mathbf{H}^{-1}$. $\mathbf{F}$ has $O(np^4)$ nonzero terms, and the number of operations is clearly bounded by $O(np^4)$. It may be less if the blocks of $\mathbf{F}$ are themselves sparse.

Term IIIa is slightly more complicated. The first part $H^{-1}_{\alpha\beta} T_{\alpha\beta\gamma}$ is comparable to term IV: each nonzero element of $\mathbf{T}$ occurs once. This takes up to $O(np^3)$ time and returns a

vector $\mathbf{v}$ with $v_\gamma = H^{-1}_{\alpha\beta} T_{\alpha\beta\gamma}$. Then term IIIa is proportional to $v_\gamma H^{-1}_{\gamma\lambda} v_\lambda = \mathbf{v}^T \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{v} = ||\mathbf{L}^{-1}\mathbf{v}||^2$. Solving $\mathbf{L}^{-1}\mathbf{v}$ takes $O(np^2)$ time since $\mathbf{L}$ is triangular and block-banded.

5. Numerical Examples and Experiments

5.1. **An Example: Poisson-Distributed Growth and the Need for Variance Stabilization.** Imagine a bacteria colony with a known rate of division per minute, θ. The deterministic equation of growth would be

$$\frac{dy}{dt} = \theta y$$

But we know that the process is not really deterministic, and that the bacteria count and number of divisions are integers. So a reasonable model is that the number of divisions over the short time $\Delta_i t$ is Poisson distributed with mean $\theta y_i \Delta_i t$:

$$\Delta_i y \sim Poiss(\theta y_i \Delta_i t)$$
$$E(\Delta_i y) = Var(\Delta_i y) = \theta y_i \Delta_i t$$

But SLAM can't handle discrete variables directly: the Laplace approximation uses an integral over the values of the y_i. The obvious fix is to use the normal approximation to the Poisson distribution:

$$\begin{aligned}
\Delta_i y &\sim \mathcal{N}(\theta y_i \Delta_i t, \theta y_i \Delta_i t) \\
\ell_{trans}(y_{i+1}|y_i) &= \frac{1}{2}\Big(log(2\pi\theta y_i \Delta_i t) + \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t}\Big) \\
\ell(\mathbf{y}) &= \sum_{i \in I_{data}} \ell_{data}(y_i|y_i^*) + \frac{1}{2}\sum_{i<N} log(2\pi\theta y_i \Delta_i t) + \frac{1}{2}\sum_{i<N} \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t}
\end{aligned}$$

where $e^{-\ell_{data}(y_i|y_i^*)}$ is the data-based likelihood of y_i given y_i^*. For now we assume that it peaks at y_i^*.

Now we should be able to find the critical path $\hat{\mathbf{y}}$ that minimizes $\ell(\mathbf{y})$, take derivatives, and compute the various Laplace terms. And presumably, for small enough time steps, this would give a good approximation to the stochastic differential equation.

Unfortunately, without a further tweak, even this simple example can fail precisely *because* of using small time steps between the data points. If there are too many consecutive time points between the data times, i.e. too many points without data, you can get the problem shown in figure 3. Increasing N tends to drive $\hat{\mathbf{y}}$ lower and lower; if there are enough points between data points, $\hat{\mathbf{y}}$ can approach zero for some time points.

This doesn't just mean that the critical paths get ugly; they are actually getting farther and farther from any realistic paths that the system might take. This is possible because the probability density of the path is large for y close to zero, even though the total

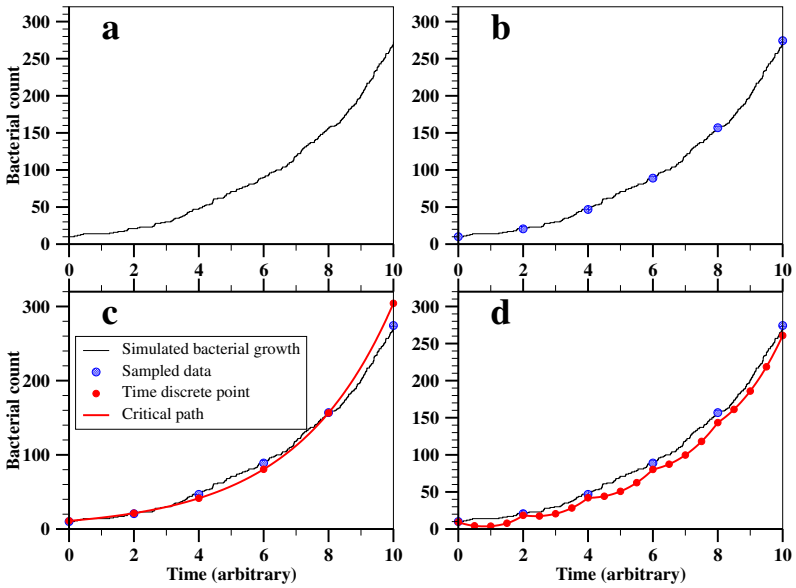


FIGURE 3. Poisson bacterial count figure.

probability measure of that region is still quite small. The problem stops being good for a Laplace-type approach.

Fortunately, there is a way around this problem. We start by identifying the reason for it, then explain the solution for this example.

The explanation below is a heuristic, not a theorem, but it is based on actual observation. Rearranging the expression for $\ell(\mathbf{y})$, we get

$$\ell(\mathbf{y}) = \frac{1}{2}\sum_{i<N} log(2\pi\Delta_i t) + \sum_{i\in I_{data}} \ell_{data}(y_i) + \frac{1}{2}\sum_{i<N} log(y_i) + \frac{1}{2}\sum_{i<N} \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t}$$

The first term is constant with respect to $\mathbf{y}$ and does not affect the minimization of $\ell(\mathbf{y})$. The next term is a data term: it only depends on y_i at the data points, and it actually increases when y_i goes below from y_i^*. So to understand the evolution of the critical path with increasing N, we have to look to the other two terms.

The third term (the $log(y)$ term) typically increases without bound as $N \to \infty$. For example, if there is a limiting path $y(t)$, with evenly spaced time points, $\sum_{i<N} log(y_i) \sim N\langle log(y)\rangle$ where $\langle log(y)\rangle$ is the mean of $log\ y(t)$.

But the last term does not increase without bound. In practice,

$$\frac{1}{2}\sum_{i<N} \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t}$$

behaves like a Riemann sum approaching its limiting value–an integral. This is because, in practice, for $\Delta_i t$ small, $\Delta_i y - \theta y_i \Delta_i t \sim O(\Delta_i t)$. (If all of the critical paths were along the same smooth function $y(t)$, then $\Delta_i y - \theta y_i \Delta_i t$ would be $O(\Delta_i t^2)$. This doesn't happen here because the y_i come from critical paths for different partitions. Essentially this sum behaves more like it would for a Brownian motion than for a smooth function.)

Using $\Delta_i y - \theta y_i \Delta_i t \sim O(\Delta_i t)$, we get

$$\begin{aligned}\sum_{i=1}^{N-1} \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t} &\sim \sum_{i=1}^{N-1} \frac{(O(\Delta_i t)^2)}{\theta y_i \Delta_i t} \\ &= \sum_{i=1}^{N-1} O(\Delta_i t)\end{aligned}$$

if the y_i are uniformly bounded below (uniformly for different N).

For this reason, as $N \to \infty$, the log-likelihood is dominated by the third term, the sum over values of *log* y_i. This is why the critical path keeps being pushed lower and lower in order to minimize ℓ.

This creates a problem: for accuracy with a continuous process, we may need small time steps, but with small time steps we get this artifact. It arises because the variance of y_{i+1} depends on y_i: a smaller y_i gives a smaller variance, which gives a greater likelihood.

The solution is to make a change of variables in the integral. If we pick the right transformation $v = v(y)$, we can stabilize the variance of v and get a realistic critical path to expand around. If $v_i = v(y_i)$, and $Var(y_i) = \theta y_i \Delta_i t$ is small,

$$\begin{aligned}Var(v_{i+1}) &\sim v'(y_i)^2 \cdot Var(y_{i+1}) \\ &= v'(y_i)^2 \cdot y_i \Delta_i t\end{aligned}$$

so we can stabilize the variance of v if

$$\begin{aligned}v'(y)^2 y &= 1 \\ v'(y)^2 &= \frac{1}{y} \\ v'(y) &= \frac{1}{\sqrt{y}} \\ v &= 2\sqrt{y}\end{aligned}$$

So $v_i = 2\sqrt{y_i}$ has stable variance. How does this affect the integral for $M(\boldsymbol{\theta})$?

If we let

$$F(\mathbf{y}) = \frac{1}{2} \sum_i \frac{(\Delta_i y - \theta y_i \Delta_i t)^2}{\theta y_i \Delta_i t}$$

then

$$L_{dyn}(\mathbf{y}) \propto \Big(\prod_i \frac{1}{\sqrt{y_i}}\Big) e^{-F(\mathbf{y})} = e^{-F(\mathbf{y})}\Big(\prod_i \frac{1}{\sqrt{y_i}}\Big)$$

and the marginalized likelihood is proportional to

$$\begin{aligned}\int L_{data}(\mathbf{y}) e^{-F(\mathbf{y})}\Big(\prod_{i<N} \frac{1}{\sqrt{y_i}}\Big) d^N\mathbf{y} &= \int L_{data}(\mathbf{y}) e^{-F(\mathbf{y})} \prod_{i<N} \frac{dy_i}{\sqrt{y_i}} \cdot dy_N \\ &= \int L_{data}(\mathbf{y}) e^{-F(\mathbf{y})} \sqrt{y_N} \prod_{i\leq N} d\ (2\sqrt{y_i}) \\ &= \int L_{data}\Big(\mathbf{y}(\mathbf{v})\Big) e^{-F(\mathbf{y}(\mathbf{v}))} \cdot \frac{v_N}{2}\ d^N\mathbf{v}\end{aligned}$$

F is always positive, so $e^{-F} \leq 1$. L_{data} is maximized when $y_i = y_i^*$. There is an additional factor of v_N, but realistically it cannot go to infinity as fast as L_{data} and e^{-F} go to zero for large v_N.

That means that this integrand, unlike the pre-transformation integrand, is bounded and has some reasonable critical path in terms of $\mathbf{v}$.

5.2. **SIR model.** Our primary model for testing comes from infectious disease epidemiology. One of the simplest widely-used models is the SIR model [Kermack and McKendrick, 1927]. There are three compartments, Susceptibles, Infected, and Recovered. At each time step, some Susceptibles become Infected, and some Infected become Recovered. Two parameters correspond to infectiousness (β) and speed of recovery (γ). The traditional method of estimating these parameters uses the deterministic equations

$$\begin{aligned}\frac{dS}{dt} &= -\beta SI \\ \frac{dI}{dt} &= \beta SI - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

The assumptions behind this are simple:

(1) new infections are proportional to contacts between susceptible and infected;

(2) new recoveries are proportional to the number infected.

Assumption (2) is memoryless, which is not very realistic, but is often adequate [Anderson and May, 1991].

To convert this to a stochastic model, we treat new infections at time i (ν_{Ii}) and new recoveries at time i (ν_{Ri}) as independent Poisson variables with means given by the deterministic model:

$$\begin{aligned}
\mu_{Ii} &= E\nu_{Ii} = \beta S_i I_i \Delta_i t \\
\nu_{Ii} &\sim Pois(\beta S_i I_i \Delta_i t) \\
\mu_{Ri} &= E\nu_{Ri} = \gamma I_i \Delta_i t \\
\nu_{Ri} &\sim Pois(\gamma I_i \Delta_i t)
\end{aligned}$$

As with the bacterial colony, we use the normal approximation:

$$\begin{aligned}
\nu_{Ii} &\sim \mathcal{N}(\mu_{Ii}, \mu_{Ii}) \\
\nu_{Ri} &\sim \mathcal{N}(\mu_{Ri}, \mu_{Ri}) \\
\Delta_i S &= -\nu_{Ii} \\
&\sim \mathcal{N}(-\mu_{Ii}, \mu_{Ii}) \\
\Delta_i I &= \nu_{Ii} - \nu_{Ri} \\
&= -\Delta_i S - \nu_{Ri} \\
&\sim \mathcal{N}(-\Delta_i S - \mu_{Ri}, \mu_{Ri})
\end{aligned}$$

We assume the measurement error is lognormal with a third parameter, σ:

$$L_{data}(\mathbf{S}, \mathbf{I}|\sigma) = \prod_{i \in I_{data}} \frac{1}{2\pi\sigma^2 S_i^* I_i^*} \exp\left(-\frac{1}{2}\frac{log(S_i/S_i^*)^2}{\sigma^2}\right) \exp\left(-\frac{1}{2}\frac{log(I_i/I^*)^2}{\sigma^2}\right)$$

The dynamical part of the likelihood is

$$L_{dyn}(\mathbf{S},\mathbf{I}|\beta,\gamma) = \prod_{i=1}^{n-1} \frac{1}{\sqrt{2\pi\mu_{Ii}}} \exp\left(-\frac{1}{2}\frac{(\Delta_i S + \mu_{Ii})^2}{\mu_{Ii}}\right) \cdot \frac{1}{\sqrt{2\pi\mu_{Ri}}} \exp\left(-\frac{1}{2}\frac{(\Delta_i I + \Delta_i S + \mu_{Ri})^2}{\mu_{Ri}}\right)$$
$$= (2\pi)^{-N/2} \prod_{i=1}^{n-1} \frac{1}{\sqrt{\mu_{Ii}\mu_{Ri}}} \exp\Big(-F(S_i, I_i, S_{i+1}, I_{i+1}|\beta,\gamma,\Delta_i t)\Big)$$

where

$$F(S_i, I_i, S_{i+1}, I_{i+1}) = \frac{1}{2}\frac{(\Delta_i S + \mu_{Ii})^2}{\mu_{Ii}} \frac{1}{2}\frac{(\Delta_i I + \Delta_i S + \mu_{Ri})^2}{\mu_{Ri}}$$

so

$$L_{dyn} = (2\pi)^{-N/2} \exp\Big(-\sum_i^{n-1} F(S_i, I_i, S_{i+1}, I_{i+1})\Big) \prod_{i=1}^{n-1} \frac{1}{\sqrt{\mu_{Ii}\mu_{Ri}}}$$
$$= (2\pi)^{-N/2} e^{-F_{all}(\mathbf{S},\mathbf{I})} \prod_{i=1}^{n-1} \frac{1}{\sqrt{\beta S_i I_i \Delta_i t \cdot \gamma I_i \Delta_i t}}$$
$$= \frac{(2\pi)^{-N/2}}{\sqrt{\beta\gamma}} e^{-F_{all}(\mathbf{S},\mathbf{I})} \prod_{i=1}^{n-1} \frac{1}{\sqrt{S_i} \cdot I_i \Delta_i t}$$

As with the bacteria model §5.1, the critical path often gets artifacts for small time steps and the approximation can be poor. So we rewrite the differential as

$$L_{dyn}\ d^n S\ d^n I \propto \exp(-F_{all}) \prod_{i=1}^{n-1} \frac{dS_i\ dI_i}{\sqrt{S_i} I_i \Delta_i t} \cdot dS_n dI_n$$
$$\propto \exp(-F_{all}) \prod_{i=1}^{n-1} \frac{dS_i}{\sqrt{S_i}} \frac{dI_i}{I_i}$$

so we want to pick transformed variables v_S and v_I such that

$$dv_S \propto \frac{dS}{\sqrt{S}},\ dv_I \propto \frac{dI}{I}$$

If we take

$$v_S = 2\sqrt{S},\ v_I = log(I)$$

then our integral avoids the artifacts explained in §5.1.

5.3. **Comparison with Rstan on British Boarding School data.** Stan [STAN Development Team] is a tool for doing Bayesian statistics using Hybrid Monte Carlo estimation. Among other things, it can estimate Bayesian posteriors for parameters in very general models, which can be specified with an easy-to-use general model language.

This makes Stan a natural check for the approximation methods in SLAM. The probabilistic model in §5.2 was defined in Stan as well as SLAM, and used to estimate parameters for a classic data set [Murray, 2002]. The purpose is to verify that the estimates comes out similar, but that SLAM can make the calculations more quickly because of the approximation. The posterior median given by Stan is not identical to the maximum marginalized likelihood estimate given by SLAM. However, for small variances, we expect them to be close.

The results are shown in the table below along with the corresponding results from SLAM. The test data set comes from an influenza infection at a British boarding school [Murray, 2002]. Initial guesses for β and γ are set equal to Murray's deterministic estimates. The initial guess for σ (the measurement error parameter) is arbitrarily set to 0.1.

Figure 4 shows the estimates of the course of infection given by SLAM, Stan, and the deterministic ODEs, given their estimated parameter values. The deterministic model suggests that the epidemic could not end as quickly as it did. Stan and SLAM can follow the data more closely using the stochastic model from §5.2. The Stan and SLAM paths are very similar, but we don't expect them to be identical because Stan is showing posterior means while SLAM is showing a overall maximum likelihood estimate.

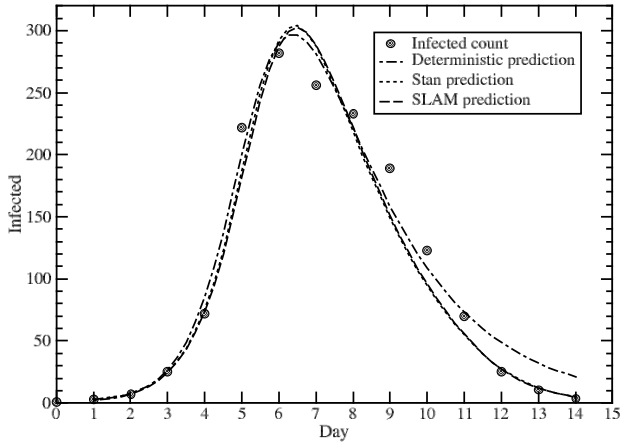


FIGURE 4. Comparison of Stan and SLAM predictions. The deterministic prediction follows the exact ODEs using the best fit values of β and γ provided by Murray. Stan shows a posterior mean, while SLAM uses a maximum likelihood path.

	β	γ	σ	Time
Deterministic (Murray)	2.18×10^{-3}	0.440	–	
Stan CI Low	2.13×10^{-3}	0.462	0.092	
Stan posterior median	2.48×10^{-3}	0.518	0.194	61.3 s
SLAM basic	2.47×10^{-3}	0.519	0.175	1.7 s
SLAM higher order	2.47×10^{-3}	0.519	0.176	4.5 s
Stan CI High	2.94×10^{-3}	0.601	0.400	

TABLE 1. SIR model results.

Table 1 shows the results of the test. As expected, the parameter estimates from Stan and SLAM are very similar. For β and γ, the differences are negligible. For σ the Stan and SLAM estimates are within a fraction of the CI, or about 10% of the value. But for this problem, compared with Stan, basic SLAM was about 35 times faster than Stan, and higher-order SLAM was more than 13 times faster.

For this test, Rstan 2.8.0 was run in R 3.2.2, using 4 chains of length 2000 each, with options set to maximize speed (multicore, optimized compilation). Different versions of Stan code for the model were tested. Surprisingly, the fastest version used unvectorized code.

5.4. **Comparison with INLA on Tokyo rain data.** INLA [Rue et al., 2009, Rue et al.] is an R package which also uses Laplace approximations to compute integrals and estimate parameters, but for a different class of problems (Gaussian Markov Random Fields [Rue and Held, 2005]). GMRFs on a line can be analyzed by both INLA and SLAM. INLA also uses sparsity, but the higher-order terms are calculated in a simpler (and more general) way which takes quadratic time and memory in the number of nodes. Therefore SLAM, with a linear-time algorithm for the higher-order terms, should be faster for sufficiently large GMRFs on a line. These experiments show that this is true for repeats of the Tokyo data set [Martino and Rue, 2009] from the INLA package.

The Tokyo data set looks at Tokyo rainfall from the start of 1983 to the end of 1984. Each row of Tokyo gives a calendar day (e.g. January 28) and the number of times it rained on that day in either 1983 or 1984. For obvious reasons this count is always 2 except for February 29 (from 1984). The top few rows are presented in Table 2.

i	n_i	y_i
1	2	0
2	2	0
3	2	1
4	2	1
5	2	0

TABLE 2. Initial rows of Tokyo rain data.

Here i is the calendar date (e.g. 1 is January 1), n_i is the number of times that date occurred, and y_i is the number of rain days for that date.

In the INLA manual [Martino and Rue, 2009], the Tokyo data set is analyzed using INLA with a second-order random walk model (Rue and Held [2005], section 3.4.1). For each date i, the chance of rain is assumed to be some p_i, so the chance of y_i rain days for that date is $\binom{n_i}{y_i} p_i^{y_i}(1-p_i)^{n_i-y_i}$. The second-order random walk is over $x_i = log(p_i/(1-p_i)$, so ℓ for given λ is

$$\ell(\mathbf{x}|\lambda) = \sum_{i=1}^{N} \Big(- y_i x_i + n_i \ log(1+e^{x_i}) \Big) + \frac{1}{2} \sum_{i=1}^{N-2} \Big(- log \ \lambda + \frac{\lambda}{\Delta_i t}(x_{i+2} - 2x_{i+1} + x_i)^2 \Big)$$

After slight modifications[1] we compared the behavior and performance of INLA and SLAM.

In order to adjust the size of the data set, we simply repeated the data set k times, e.g., for k=2 there are 732 data points instead of 366 and $y_{i+366} = y_i$.

By default, INLA uses an approximation to the higher order terms which is easier to compute (strategy='simplified.laplace'). To get the full higher order Laplace terms, as used by SLAM, we set INLA strategy = 'laplace'.

After running both packages on Tokyo, the estimates are not identical but are very similar, especially for the larger data sets (see figures). The most important results here have to do with run time and especially memory use. For k up to 3, INLA is faster. At k=4 ($N \sim 1.5 \times 10^3$), SLAM becomes faster. The relative difference grows as shown in figure 6.

The SSE for the simplified Laplace is several times larger than for INLA full or for SLAM. This is almost entirely due to greater error near the ends of the year (1 and 366).

The expected number of rain days over the year is $\sum_i n_i \hat{p}_i$ and the observed number of rain days is $\sum_i y_i$. For SLAM these are equal to at least four digits. This probably means that, under some conditions, the method forces them to be equal. For now we make no suggestion why.

Lastly, in this example we had to introduce a variable $\hat{x}'$ which is supposed to be the time derivative of the linear predictor x. We forced $\hat{x}'$ to be close to the true derivative by adding a large penalty for any difference between $\hat{x}'_i$ and $\Delta_i x/\Delta_i t$. These results show that it is (sometimes) possible to use SLAM for a constrained system using this simple trick.

[1]Three modifications were made in order to work around features that are not yet present in the SLAM code.

1. SLAM doesn't take second derivatives, so an extra variable $\hat{x}'_i$ is added which is forced to be nearly equal to the true $x'(t_i)$. Then the usual RW2 penalty is given in terms of $\sum_i (\Delta_i \hat{x}')^2$.

2. INLA has an option "cyclic" which forces the function to be periodic. SLAM does not currently have such an option. Therefore cyclic is set to FALSE for INLA. Happily, the predicted functions are nearly periodic.

3. The Tokyo data set covers 1983-84, so $n_i = 2$ for all dates except February 29. For Feb. 29 $n_i = 1$ because only 1984 was a leap year. For now, it was necessary to set $n_i = 2$ for February 29 for both programs. Think of it as a recording error.

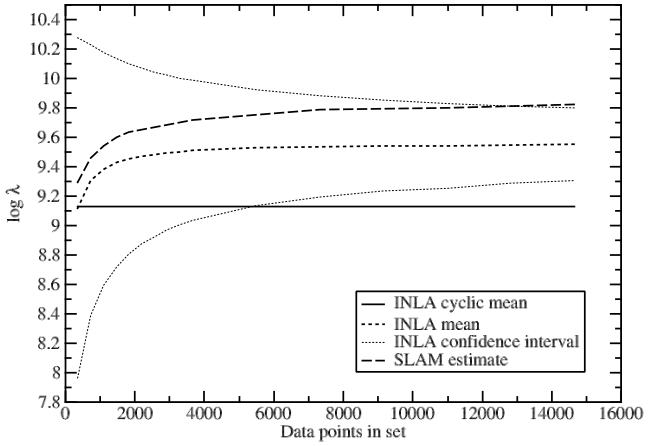


FIGURE 5. Comparison of INLA estimates, the means and ranges.

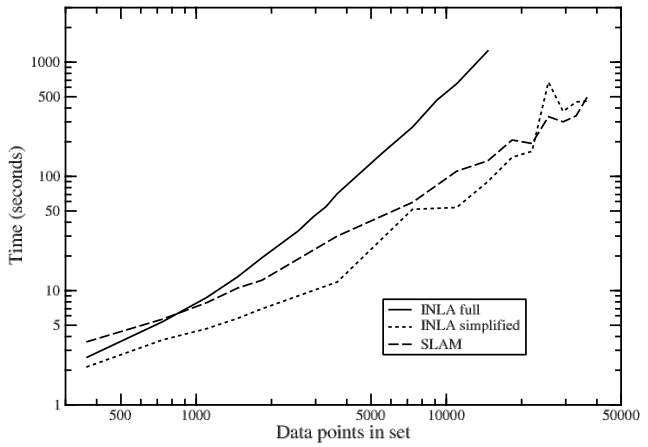


FIGURE 6. Comparison of INLA and SLAM run times.

6. Discussion

We have shown how to efficiently compute a higher-order Laplace approximation for functional integrals on time. This can be used to define a marginalized pseudolikelihood for differential equation systems involving randomness. In turn, these approximate marginalized likelihoods can be maximized in order to estimate parameters of the system, such as

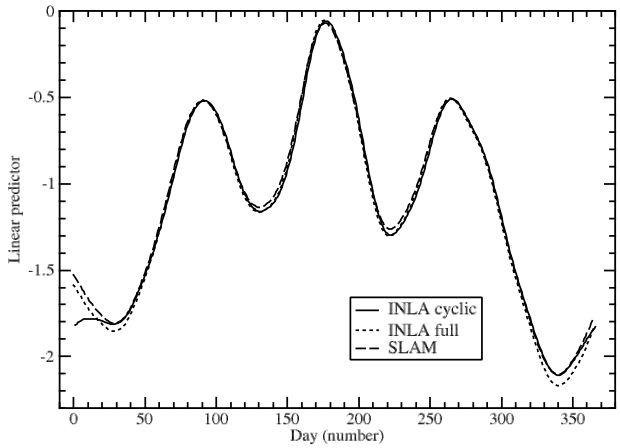


FIGURE 7. Linear predictors.

infectiousness of a disease, or a good roughness parameter for a smoothing problem. This approximation is implemented in the MATLAB package SLAM. SLAM is compared on real data against two leading packages which also estimate/predict parameters according to a fully Bayesian system or mixed model. As predicted, the time needed for SLAM seems to grow linearly. Compared with either package, for medium-sized data sets using higher-order terms, SLAM produces equivalent results in far less time. This is especially interesting given that both Stan and INLA use compiled, optimized code. Finally, for some problems, if higher-order terms are used then SLAM is able to process significantly larger data sets than INLA.

7. Acknowledgements

This work was begun at the Center on Aging and Health, Johns Hopkins Bloomberg School of Public Health, under Training Grant T32AG000247 (Epidemiology and Biostatistics of Aging). Additional support was provided by the Biostatistics Department at JHSPH. I would like to thank Alan Cohen (now at Sherbrooke University) for inviting me into the project which led to this. I would also like to thank Ravi Varadhan and Giles Hooker for consultation and encouragement, and Karen Bandeen-Roche for making it possible. Michael Schumaker provided help with editing and LaTeX. Thanks are also due to the developers of Stan, INLA, and the MATLAB Tensor Toolbox for quick and friendly answers to my questions.

8. Appendix: Calculation of Term IIIb in linear time

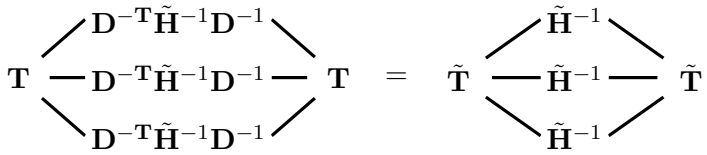
8.1. **Tensor levels.** This shows how to compute term IIIb in $O(np^4)$ time. This is possible because of the raising and lowering principle described in section 4. A similar concept

applies to block tensors. A tensor of order 3, $\mathbf{X}$, is block off-diagonal with level (l_1, l_2, l_3) if every nonzero block $\mathbf{X_{i_1,i_2,i_3}}$ has $i_1 - l_1 = i_2 - l_2 = i_3 - l_3$. In other words, $\mathbf{X}$ is level (l_1, l_2, l_3) if it is block off-diagonal and includes the block (l_1, l_2, l_3). Levels are an equivalence class: level (l_1, l_2, l_3) is the same as level $(l_1 + \Delta l, l_2 + \Delta l, l_3 + \Delta l)$ for any integer Δl.

The levels of a third-order block tensor are shifted when powers of $\mathbf{A}$ or $\mathbf{A^T}$ are applied to the modes. Our $\mathbf{T}$ is a sum of a handful of block-diagonal and block off-diagonal levels. To compute IIIb, we find the combinations of powers of $\mathbf{A}$ and $\mathbf{A^T}$ which connect the levels of the first $\mathbf{T}$ with the levels of the second $\mathbf{T}$. The grouping is more complicated than for computing the near-diagonals of $\mathbf{H}^{-1}$, but is similar in spirit.

Suppose have a single-level matrix $\mathbf{B}$ and a single-level tensor $\mathbf{T_{mono}}$ with level (l_1, l_2, l_3). If $\mathbf{B}$ is level l_B, then the new tensor $(\mathbf{B} \otimes \mathbf{I} \otimes \mathbf{I})(\mathbf{T_{mono}})$ is also block off-diagonal, but with level $(l_1 + l_B, l_2, l_3)$. The next observation is critical: if $\mathbf{B}$ is applied to every mode of $\mathbf{T_{mono}}$, the result $(\mathbf{B} \otimes \mathbf{B} \otimes \mathbf{B})\,(\mathbf{T_{mono}})$ is the same exact level as $\mathbf{T_{mono}}$.

8.2. **A simplifying coordinate transformation.** As we discussed in section 8, we can write term IIIb more simply by making a coordinate transformation:



where

$$\tilde{\mathbf{T}} = (\mathbf{D^{-T}} \otimes \mathbf{D^{-T}} \otimes \mathbf{D^{-T}})(\mathbf{T}).$$

$\mathbf{D}$ is block-diagonal with block size p, so the coordinate transformation takes $O(np^4)$ time.

To continue, we will use the series

$$\tilde{\mathbf{H}}^{-1} = \mathbf{S} \sum_{\mu>0} \mathbf{A}^{\boldsymbol{\mu}} + \mathbf{S} + \sum_{\nu>0} \mathbf{A^{T\nu}} \mathbf{S}$$

We simplify this by making a second block-diagonal coordinate transformation to eliminate $\mathbf{S}$. Note that $\mathbf{S}$ is block-diagonal and symmetric because each of its terms $\mathbf{A^{Tq}A^q}$ is block-diagonal and symmetric. $\mathbf{S}$ is also positive definite, since $\mathbf{I}$ is positive definite and every $\mathbf{A^{Tq}A^q}$ is semidefinite. So if we define $\mathbf{C}$ to be the (upper!) Cholesky decomposition $\mathbf{S} = \mathbf{C^T C}$. Then $\mathbf{C^{-T}S} = \mathbf{C}$, $\mathbf{SC^{-1}} = \mathbf{C^T}$ and $\mathbf{C^{-T}SC^{-1}} = \mathbf{I}$. Now we can make the coordinate transformation

$$\mathbf{y} \to \mathbf{C}^{-1}\mathbf{y}, \quad \hat{\mathbf{H}} = \mathbf{C^T}\tilde{\mathbf{H}}\mathbf{C}$$

from which

$$\begin{aligned}\hat{\mathbf{H}}^{-\mathbf{1}} &= \mathbf{C}^{-\mathbf{T}}\tilde{\mathbf{H}}^{-\mathbf{1}}\mathbf{C}^{-\mathbf{1}} \\ &= \mathbf{C}^{-\mathbf{T}}\Big(\mathbf{S}\sum_{\mu>0}\mathbf{A}^{\boldsymbol{\mu}} + \mathbf{S} + \sum_{\nu>0}\mathbf{A}^{\mathbf{T}\boldsymbol{\nu}}\mathbf{S}\Big)\mathbf{C}^{-1} \\ &= \mathbf{C}^{-\mathbf{T}}\Big(\mathbf{S}\sum_{\mu>0}\mathbf{A}^{\boldsymbol{\mu}}\Big)\mathbf{C}^{-1} + \mathbf{I} + \mathbf{C}^{-\mathbf{T}}\Big(\sum_{\nu>0}\mathbf{A}^{\mathbf{T}\boldsymbol{\nu}}\mathbf{S}\Big)\mathbf{C}^{-1}\end{aligned}$$

The first and third terms are transposes of each other. Let's look at the first term alone.

$$\begin{aligned}\mathbf{C}^{-\mathbf{T}}\Big(\mathbf{S}\sum_{\mu>0}\mathbf{A}^{\boldsymbol{\mu}}\Big)\mathbf{C}^{-1} &= \mathbf{C}\Big(\sum_{\mu>0}\mathbf{A}^{\boldsymbol{\mu}}\Big)\mathbf{C}^{-1} \\ &= \sum_{\mu>0}\Big(\mathbf{C}\mathbf{A}^{\boldsymbol{\mu}}\mathbf{C}^{-1}\Big) \\ &= \sum_{\mu>0}\hat{\mathbf{A}}^{\boldsymbol{\mu}}\end{aligned}$$

with $\hat{\mathbf{A}} = \mathbf{C}^{-\mathbf{1}}\mathbf{A}\mathbf{C}$. At last,

$$\begin{aligned}\hat{\mathbf{H}}^{-\mathbf{1}} &= \sum_{\mu>0}\hat{\mathbf{A}}^{\boldsymbol{\mu}} + \mathbf{I} + \sum_{\nu>0}\hat{\mathbf{A}}^{T\boldsymbol{\nu}} \\ &= \sum_{\mu\geq 0}\hat{\mathbf{A}}^{\boldsymbol{\mu}} + \sum_{\nu\geq 0}\hat{\mathbf{A}}^{T\boldsymbol{\nu}} - \mathbf{I} \\ &= \mathcal{A} + \mathcal{A}^{\mathbf{T}} - \mathbf{I}\end{aligned}$$

where $\mathcal{A} = \sum_{\mu\geq 0}\hat{\mathbf{A}}$.

8.3. **Statement of Results.**

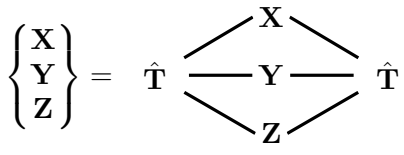
$$\begin{aligned}IIIb = 2\langle\hat{\mathbf{T}}, \mathcal{S}(\hat{\mathbf{T}})\rangle &\\ + 6\langle\hat{\mathbf{T}}, \Big(\mathbf{I}\otimes(\mathbf{I}+\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\otimes(\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\Big)(\mathcal{S}(\hat{\mathbf{T}}))\rangle &\\ + 6\langle\hat{\mathbf{T}}, \Big((\mathbf{I}+\hat{\mathbf{A}})\otimes\hat{\mathbf{A}}\otimes\hat{\mathbf{A}}^{\mathbf{T}}\Big)(\hat{\mathbf{T}})\rangle &\\ - \langle\hat{\mathbf{T}}, \hat{\mathbf{T}}\rangle &\end{aligned}$$

where $\langle\mathbf{P}, \mathbf{Q}\rangle = P_{\alpha\beta\gamma}Q_{\alpha\beta\gamma}$
and $\mathcal{S}(\hat{\mathbf{T}})$ is defined as

$$\mathcal{S}(\hat{\mathbf{T}}) = \sum_{q\geq 0}\Big(\hat{\mathbf{A}}^{\mathbf{q}}\otimes\hat{\mathbf{A}}^{\mathbf{q}}\otimes\hat{\mathbf{A}}^{\mathbf{q}}\Big)(\hat{\mathbf{T}}).$$

$\mathcal{S}(\hat{\mathbf{T}})$ can be computed in $O(np^4)$ time, and so can the whole sum.

8.4. **How to expand the sum and group the terms.** Let's start by defining a more compact notation.



This has the properties of multilinearity and symmetry under permutation:

$$\left\{\begin{matrix}\mathbf{W}+\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\} = \left\{\begin{matrix}\mathbf{W}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\} + \left\{\begin{matrix}\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\}$$

$$\left\{\begin{matrix}\lambda\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\} = \lambda \left\{\begin{matrix}\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\}$$

$$\left\{\begin{matrix}\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\} = \left\{\begin{matrix}\mathbf{Y}\\ \mathbf{X}\\ \mathbf{Z}\end{matrix}\right\} = \left\{\begin{matrix}\mathbf{Y}\\ \mathbf{Z}\\ \mathbf{X}\end{matrix}\right\}$$

Furthermore, if *all* the matrices are transposed, then you get the same value:

$$\left\{\begin{matrix}\mathbf{X}\\ \mathbf{Y}\\ \mathbf{Z}\end{matrix}\right\} = \left\{\begin{matrix}\mathbf{X}^\mathbf{T}\\ \mathbf{Y}^\mathbf{T}\\ \mathbf{Z}^\mathbf{T}\end{matrix}\right\}$$

Multilinearity and permutation symmetry mean that we can expand our bracket of sums in the same way as with a product of sums.

$$\begin{Bmatrix} \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \\ \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \\ \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} = \begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A} \end{Bmatrix} + \begin{Bmatrix} \mathcal{A}^{\mathbf{T}} \\ \mathcal{A}^{\mathbf{T}} \\ \mathcal{A}^{\mathbf{T}} \end{Bmatrix}$$
$$+ 3\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A}^{\mathbf{T}} \end{Bmatrix} + 3\begin{Bmatrix} \mathcal{A}^{\mathbf{T}} \\ \mathcal{A}^{\mathbf{T}} \\ \mathcal{A} \end{Bmatrix}$$
$$- 3\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathbf{I} \end{Bmatrix} - 3\begin{Bmatrix} \mathcal{A}^{\mathbf{T}} \\ \mathcal{A}^{\mathbf{T}} \\ \mathbf{I} \end{Bmatrix} - 6\begin{Bmatrix} \mathcal{A} \\ \mathbf{I} \\ \mathcal{A}^{\mathbf{T}} \end{Bmatrix}$$
$$+ 3\begin{Bmatrix} \mathcal{A} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix} + 3\begin{Bmatrix} \mathcal{A}^{\mathbf{T}} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$
$$- \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$

Combining transposes lets us group the terms together further:

$$\begin{Bmatrix} \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \\ \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \\ \mathcal{A}+\mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} = 2\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A} \end{Bmatrix}$$
$$+ 6\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A}^{\mathbf{T}} \end{Bmatrix} - 6\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathbf{I} \end{Bmatrix} - 6\begin{Bmatrix} \mathcal{A} \\ \mathbf{I} \\ \mathcal{A}^{\mathbf{T}} \end{Bmatrix} + 6\begin{Bmatrix} \mathcal{A} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$
$$- \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$
$$= 2\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A} \end{Bmatrix} + 6\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} - 6\begin{Bmatrix} \mathcal{A} \\ \mathbf{I} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} - \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$
$$= 2\begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A} \end{Bmatrix} + 6\begin{Bmatrix} \mathcal{A} \\ \mathcal{A}-\mathbf{I} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} - \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$$

The calculation of $\begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}$ is trivial: it's just $\langle \hat{\mathbf{T}}, \hat{\mathbf{T}} \rangle$. The other two involve grouping the tiny minority of *nonzero* subterms in the first two terms. The principle to remember is that none of the nonzero levels are more than two apart in any of the 3 modes. This means

that $\begin{Bmatrix} \hat{\mathbf{A}}^{\mathbf{q}} \\ \hat{\mathbf{A}}^{\mathbf{r}} \\ \hat{\mathbf{A}}^{\mathbf{s}} \end{Bmatrix} = 0$ if any of $|q-r|, |q-s|, |r-s|$ are more than 2, and $\begin{Bmatrix} \hat{\mathbf{A}}^{\mathbf{q}} \\ \hat{\mathbf{A}}^{\mathbf{r}} \\ \hat{\mathbf{A}}^{\mathbf{Ts}} \end{Bmatrix} = 0$ if either $q+s$ or $r+s$ is greater than 2. Taking the middle term first,

$$\begin{aligned} \begin{Bmatrix} \mathcal{A} \\ \mathcal{A}-\mathbf{I} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} &= \begin{Bmatrix} \mathbf{I}+\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}}+... \\ \hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}}+... \\ \hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{T2}}+... \end{Bmatrix} \\ &= \begin{Bmatrix} \mathbf{I}+\hat{\mathbf{A}} \\ \hat{\mathbf{A}} \\ \hat{\mathbf{A}}^{\mathbf{T}} \end{Bmatrix} \end{aligned}$$

$$\begin{aligned} \begin{Bmatrix} \mathcal{A} \\ \mathcal{A}-\mathbf{I} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} &= \begin{Bmatrix} \sum_{q\geq 0} \hat{\mathbf{A}}^{\mathbf{q}} \\ \sum_{r>0} \hat{\mathbf{A}}^{\mathbf{r}} \\ \sum_{s>0} \hat{\mathbf{A}}^{\mathbf{Ts}} \end{Bmatrix} \\ &= \sum_{q\geq 0}\sum_{r>0}\sum_{s>0} \begin{Bmatrix} \hat{\mathbf{A}}^{\mathbf{q}} \\ \hat{\mathbf{A}}^{\mathbf{r}} \\ \hat{\mathbf{A}}^{\mathbf{Ts}} \end{Bmatrix} \\ &= \sum_{s>0}\sum_{q=0}^{2-s}\sum_{r=1}^{2-s} \begin{Bmatrix} \hat{\mathbf{A}}^{\mathbf{q}} \\ \hat{\mathbf{A}}^{\mathbf{r}} \\ \hat{\mathbf{A}}^{\mathbf{Ts}} \end{Bmatrix} \end{aligned}$$

All terms have power $r, s \geq 1$. But this implies that the only nonzero terms have $q, r, s \leq 1$. If q were greater than 1, we would have $q+s > 1+s \geq 2$. $r \leq 1$ for similar reasons. And if s were greater than 1 we would have $s+r > 1+r \geq 2$. Consequently, this triply-infinite sum reduces to

$$\begin{Bmatrix} \mathcal{A} \\ \mathcal{A}-\mathbf{I} \\ \mathcal{A}^{\mathbf{T}}-\mathbf{I} \end{Bmatrix} = \sum_{q=0}^{1}\sum_{r=1}^{1}\sum_{s=1}^{1} \begin{Bmatrix} \hat{\mathbf{A}}^{\mathbf{q}} \\ \hat{\mathbf{A}}^{\mathbf{r}} \\ \hat{\mathbf{A}}^{\mathbf{Ts}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{I}+\hat{\mathbf{A}} \\ \hat{\mathbf{A}} \\ \hat{\mathbf{A}}^{\mathbf{T}} \end{Bmatrix}.$$

Finally, we reach the first sum, which is also the most complicated:

$$
\begin{aligned}
\begin{Bmatrix}\mathcal{A}\\ \mathcal{A}\\ \mathcal{A}\end{Bmatrix} &= \sum_{q\geq 0}\sum_{r\geq 0}\sum_{s\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{r}}\\ \hat{\mathbf{A}}^{\mathbf{s}}\end{Bmatrix}\\
&= \sum_{q=r=s}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{r}}\\ \hat{\mathbf{A}}^{\mathbf{s}}\end{Bmatrix} + 3\sum_{q=r<s}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{r}}\\ \hat{\mathbf{A}}^{\mathbf{s}}\end{Bmatrix} + 3\sum_{q<r=s}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{r}}\\ \hat{\mathbf{A}}^{\mathbf{s}}\end{Bmatrix} + 6\sum_{q<r<s}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{r}}\\ \hat{\mathbf{A}}^{\mathbf{s}}\end{Bmatrix}\\
&= \sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} + 3\sum_{q\geq 0}\left(\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}\end{Bmatrix} + \begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix}\right) + 3\sum_{q\geq 0}\left(\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}\end{Bmatrix} + \begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+2}}\\ \hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix}\right) + 6\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}\\ \hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix}\\
&= \sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} + 3\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}+\hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix} + 3\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}+\hat{\mathbf{A}}^{\mathbf{q+2}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}+\hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix}\\
&= \sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} + 3\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}+\hat{\mathbf{A}}^{\mathbf{q+1}}+\hat{\mathbf{A}}^{\mathbf{q+2}}\\ \hat{\mathbf{A}}^{\mathbf{q+1}}+\hat{\mathbf{A}}^{\mathbf{q+2}}\end{Bmatrix}\\
&= \sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} + 3\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ (\mathbf{I}+\hat{\mathbf{A}}+\hat{\mathbf{A}}^2)\hat{\mathbf{A}}^{\mathbf{q}}\\ (\hat{\mathbf{A}}+\hat{\mathbf{A}}^2)\hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix}
\end{aligned}
$$

Using

$$\mathcal{S}(\hat{\mathbf{T}}) = \sum_{q\geq 0}\left(\hat{\mathbf{A}}^{\mathbf{q}}\otimes\hat{\mathbf{A}}^{\mathbf{q}}\otimes\hat{\mathbf{A}}^{\mathbf{q}}\right)(\hat{\mathbf{T}})$$

then these terms become

$$\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\\ \hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} = \langle\hat{\mathbf{T}}, \mathcal{S}(\hat{\mathbf{T}})\rangle$$

$$\sum_{q\geq 0}\begin{Bmatrix}\hat{\mathbf{A}}^{\mathbf{q}}\\ (\mathbf{I}+\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\hat{\mathbf{A}}^{\mathbf{q}}\\ (\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\hat{\mathbf{A}}^{\mathbf{q}}\end{Bmatrix} = \langle\hat{\mathbf{T}}, \left(\mathbf{I}\otimes(\mathbf{I}+\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\otimes(\hat{\mathbf{A}}+\hat{\mathbf{A}}^{\mathbf{2}})\right)(\mathcal{S}(\hat{\mathbf{T}}))\rangle$$

Putting the whole expression together,

$$
\begin{aligned}
IIIb &= \begin{Bmatrix} \mathcal{A} + \mathcal{A}^{\mathbf{T}} - \mathbf{I} \\ \mathcal{A} + \mathcal{A}^{\mathbf{T}} - \mathbf{I} \\ \mathcal{A} + \mathcal{A}^{\mathbf{T}} - \mathbf{I} \end{Bmatrix} \\
&= 2 \begin{Bmatrix} \mathcal{A} \\ \mathcal{A} \\ \mathcal{A} \end{Bmatrix} + 6 \begin{Bmatrix} \mathcal{A} \\ \mathcal{A} - \mathbf{I} \\ \mathcal{A}^{\mathbf{T}} - \mathbf{I} \end{Bmatrix} - \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix} \\
&= 2\Big(\langle \hat{\mathbf{T}}, \mathcal{S}(\hat{\mathbf{T}}) \rangle + 3\langle \hat{\mathbf{T}}, \Big(\mathbf{I} \otimes (\mathbf{I} + \hat{\mathbf{A}} + \hat{\mathbf{A}}^{\mathbf{2}}) \otimes (\hat{\mathbf{A}} + \hat{\mathbf{A}}^{\mathbf{2}})\Big)(\mathcal{S}(\hat{\mathbf{T}})) \rangle\Big) + 6 \begin{Bmatrix} \mathbf{I} + \hat{\mathbf{A}} \\ \hat{\mathbf{A}} \\ \hat{\mathbf{A}}^{\mathbf{T}} \end{Bmatrix} - \begin{Bmatrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \end{Bmatrix}
\end{aligned}
$$

Like the computation in near-diags there is a simple, linear recurrence relation for $\mathcal{S}(\hat{\mathbf{T}})$. First, $\mathcal{S}(\hat{\mathbf{T}})$ solves a linear equation:

$$
\begin{aligned}
\mathcal{S}(\hat{\mathbf{T}}) &= \hat{\mathbf{T}} + \sum_{q>0} \Big(\hat{\mathbf{A}}^{\mathbf{q}} \otimes \hat{\mathbf{A}}^{\mathbf{q}} \otimes \hat{\mathbf{A}}^{\mathbf{q}}\Big)(\hat{\mathbf{T}}) \\
&= \hat{\mathbf{T}} + \Big(\hat{\mathbf{A}} \otimes \hat{\mathbf{A}} \otimes \hat{\mathbf{A}}\Big)(\mathcal{S}(\hat{\mathbf{T}}))
\end{aligned}
$$

In block notation, this gives us a recurrence relation similar to **??**. This is a forward recurrence instead of backwards because of

$$
\begin{aligned}
\mathcal{S}(\hat{\mathbf{T}})_{ijk} &= \hat{\mathbf{T}}_{ijk} + \sum_{i'j'k'} \Big(\hat{\mathbf{A}}_{ii'} \otimes \hat{\mathbf{A}}_{jj'} \otimes \hat{\mathbf{A}}_{kk'}\Big)(\mathcal{S}(\hat{\mathbf{T}})_{i'j'k'}) \\
&= \hat{\mathbf{T}}_{ijk} \text{ if i=1 or j=1 or k=1} \\
&= \hat{\mathbf{T}}_{ijk} + \Big(\hat{\mathbf{A}}_{i,i-1} \otimes \hat{\mathbf{A}}_{j,j-1} \otimes \hat{\mathbf{A}}_{k,k-1}\Big)(\mathcal{S}(\hat{\mathbf{T}})_{i-1,j-1,k-1}) \text{ otherwise}
\end{aligned}
$$

The recurrence is forward instead of backward because we chose to use $\sum_{q>0} \Big(\hat{\mathbf{A}}^{\mathbf{q}} \otimes \hat{\mathbf{A}}^{\mathbf{q}} \otimes \hat{\mathbf{A}}^{\mathbf{q}}\Big)(\hat{\mathbf{T}})$ rather than $\sum_{q>0} \Big(\hat{\mathbf{A}}^{\mathbf{Tq}} \otimes \hat{\mathbf{A}}^{\mathbf{Tq}} \otimes \hat{\mathbf{A}}^{\mathbf{Tq}}\Big)(\hat{\mathbf{T}})$ for the main sum. Each successive block of $\mathcal{S}(\hat{\mathbf{T}})$ is computed in terms of a known block of $\hat{\mathbf{T}}$ and (if present) the previous block of $\mathcal{S}(\hat{\mathbf{T}})$ from the same level. There are 7 block levels, the number of blocks per level is $O(n)$, and the number of operations in a matrix-times-block operation is $O(p^4)$, so the complexity of finding $\mathcal{S}(\hat{\mathbf{T}})$ is $O(np^4)$.

The computation can be accelerated by separating the levels of $\hat{\mathbf{T}}$ and making greater use of symmetry, but it will still be $O(np^4)$.

References

R. Anderson and R. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, Oxford, UK, 1991.

A. Asif and J.M.F. Moura. Block matrices with L-block-banded inverse: inversion algorithms. *IEEE Transactions on Signal Processing*, 53(2):630–642, 2005.

D. Campbell and R.J. Steele. Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, 22(2):429–443, 2012.

P. Dirac. The Lagrangian in quantum mechanics. *Physikalische Zeitschrift der Sowjetunion*, 3:64–72, 1933.

L. Edelstein-Keshet. *Mathematical Models in Biology*. Random House, New York, NY, USA, 1988.

R. P. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. Dover, 2010. (Original published in 1965).

A. Gelman, F. Bois, and J. Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *J. Am. Statist. Ass.*, 91(436):1400–1412, 1996.

W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proc. Roy. Soc. Lond. A*, 115(772):700–721, 1927.

H. Kleinert. *Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets*. World Scientific, 5th edition, 2009. Individual chapters available at http://users.physik.fu-berlin.de/ kleinert.

H Lineweaver and D. Burk. The determination of enzyme dissociation constants. *Journal of the American Chemical Society*, 56(3), 1934.

S. Martino and H. Rue. *INLA manual*. 2009. Available at http://www.math.ntnu.no/ hrue/GMRFsim/manual.pdf.

P. McCullagh. *Tensor Methods in Statistics*. Chapman & Hall, London, UK, and New York, NY, USA, 1987.

J.D. Murray. *Mathematical Biology I, An Introduction*. Springer-Verlag, 2002.

Hooker G.J. Ramsay J. et al. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society B*, 69:741–796, 2007.

D. Romer. *Advanced Macroeconomics*. McGraw-Hill, Columbus, OH, USA, 4th edition, 2011.

H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, 2005.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J. R. Statist. Soc. B*, 71(2): 319–392, 2009.

H. Rue et al. The r-inla project. `http://www.r-inla.org`.

L. S. Schulman. *Techniques and Applications of Path Integration*. Dover, 2005. Reprint of 1981 version.

Z. Shun and P. McCullagh. Laplace Approximation of high dimensional integrals. *J. R. Statist. Soc. B*, 57(4):749–760, 1995.

STAN Development Team. *STAN: A C++ Library for Probability and Sampling, Version 2.1*. STAN, 2013. http://mc-stan.org.